Conseils d'utilisation de STAT_NON_LINE

Résumé

L'objectif de cette note est de donner quelques conseils à un utilisateur souhaitant réaliser des calculs non linéaires avec *Code_Aster* en utilisant l'opérateur STAT_NON_LINE [U4.51.03].

Des conseils généraux sont d'abord donnés. Puis on précise les solutions à mettre en œuvre pour les principaux types de problèmes rencontrés. Les problèmes dont il est question sont de type non-convergence, ou échec d'un algorithme. On renvoie également vers la documentation spécifique à chaque problème.

Table des matières

1 Introduction

Ce document a pour but d'aider l'utilisateur à résoudre un problème d'évolution mécanique ou thermo-hydro-mécanique, en quasi-statique, d'une structure en non-linéaire, *via* l'opérateur STAT_NON_LINE. Une présentation détaillée de l'algorithme de cet opérateur est faite dans sa documentation théorique [R5.03.01].
Les 3 grands types de non linéarités sont les suivantes :

- non-linéarité liée au comportement du matériau (par exemple plastique) ;
- non-linéarité liée à la géométrie (par exemple en grands déplacements) ;
- non-linéarité liée au contact-frottant.

Dans la 2ème partie de ce document, on détaille les principaux outils dont on dispose pour aider la convergence des calculs et améliorer la qualité des résultats. Les parties suivantes sont consacrées aux conseils d'utilisation. Dans la 3ème partie, on donne des conseils généraux, communs aux 3 types de non-linéarités. Dans la 4ème, 5ème et 6ème parties, on donne des conseils dédiés à chacun des 3 types de non-linéarités : matériaux, puis géométrique et enfin de contact-frottant. Ces conseils font référence aux outils évoqués dans en 2ème partie.

2 Outils d'aide à la convergence

En cas de problèmes, on dispose de plusieurs outils :

- sous-découpage du pas de temps,
- changer les matrices,
- pilotage,
- recherche linéaire,
- changer de formulation éléments finis.

2.1 Le sous-découpage du pas de temps

De manière générale, plus le pas de temps est petit, moins le problème est non-linéaire, donc plus facile à résoudre. Le sous-découpage du pas de temps est donc un outil essentiel qui permet de passer les difficultés les plus courantes. Le sous-découpage s'active par défaut dans DEFI_LIST_INST [U4.34.03]. On conseille de toujours l'activer. En outre, plus le pas de temps est petit, et plus l'erreur de discrétisation temporelle est petite.

2.2 Le choix judicieux des matrices

Dans l'algorithme de Newton, il est possible d'utiliser différentes matrices en prédiction et en correction et de les ré-actualiser plus ou moins souvent. Ces choix sont spécifiés dans STAT_NON_LINE sous le mot-clé NEWTON [U4.51.03]. Le choix le plus polyvalent et robuste est d'utiliser la matrice tangente ré-actualisée à chaque itération de Newton et de choisir une prédiction élastique.

```
PREDICTION='ELASTIQUE'
MATRICE='TANGENTE'  (par défaut)
REAC_ITER=1
REAC_INCR=1 (par défaut)
```

En cas de problème de convergence pour passer un instant délicat, il peut être intéressant de basculer de la matrice tangente vers la matrice de décahrge si le pas de temps devient trop petit (c'est-à-dire lorsque le pas de temps en question a été sous-découpé de nombreuses fois successivement). La valeur du pas de temps en dessous de laquelle on prend la matrice de décharge est donnée par le mot-clé PAS_MINI_ELAS. La fréquence d'actualisation de la matrice de décharge est définie par REAC_ITER_ELAS. Pour une définition précise de la matrice de décharge, voir le mot-clé facteur NEWTON de [U4.51.03]).

2.3 Le pilotage

Le pilotage du chargement est une méthode de continuation pour la méthode de Newton. Le pilotage permet en particulier de calculer la réponse d'une structure qui présenterait des instabilités, aussi bien d'origines géométrique (flambement) que matériau (adoucissement). Son emploi est limité à des simulations pour lesquelles le temps ne joue pas de rôle physique, ce qui exclut *a priori* les problèmes dynamiques, ou visqueux ou thermo-mécanique.

2.4 La recherche linéaire

La méthode de Newton fournit un incrément des inconnues, mais cet incrément n'est valide que dans un voisinage de l'initialisation. L'idée de la recherche linéaire est d'utiliser la direction de l'incrément, mais en contrôlant la longueur d'avancée dans cette direction. Le pas d'avancement est alors choisi par minimisation d'une fonctionnelle. Cela permet notamment d'éviter certaines divergences de l'algorithme de Newton. Cependant, l'activation de la recherche linéaire coûte « cher ». Il est recommandé de ne l'activer qu'en cas de besoin.

2.5 Le choix judicieux de la formulation éléments-finis

Tous les éléments-finis ne se valent pas. Certains éléments se comportent mieux que d'autres dans certaines situations [U2.01.10]. Le choix des éléments-finis est fait dans l'opérateur `AFFE_MODELE`.

3 Conseils communs aux 3 types de non-linéarités

Dans cette partie, on donne des conseils généraux sur la résolution de problèmes non-linéaires

3.1 Gestion de la liste d'instants

La gestion de la liste d'instants est spécifiée via l'opérateur `DEFI_LIST_INST` [U4.34.03]. Par défaut, le sous-découpage de la liste d'instants est activé en cas d'erreur (dans le contact, la loi de comportement, le pilotage, la factorisation...). Il est également possible de déclencher le sous-découpage du pas de temps lorsque l'incrément d'une grandeur dépasse un seuil fixé (`DELTA_GRANDEUR`). Cela permet par exemple de s'assurer qu'une variable interne ne varie pas trop au cours d'un pas temps. Une autre cause possible de sous-découpage est une non-diminution du résidu au cours des itérations de Newton (`DIVE_RESI`). Cela permet de sous-découper en cas de problème sans attendre d'arriver au bout du nombre d'itérations de Newton autorisées.
La gestion dite manuelle de la liste d'instants est la gestion classique : le calcul suit la liste d'instants fournie par l'utilisateur (avec possibilité de sous-découpage). Il est également possible d'opter pour une gestion automatique de la liste d'instants. Dans ce cas, l'utilisateur ne donne que quelques instants de passages (instants où le chargement change par exemple, ou instants de post-traitement).
D'après un premier retour d'expérience, la gestion automatique du pas de temps [bib1] :

- possède un réel intérêt si l'utilisateur n'a aucune connaissance *a priori* de la discrétisation temporelle,
- permet souvent de gagner un temps considérable pour les études « faiblement » non-linéaires, c'est-à-dire les études qui convergent sans techniques avancées (pilotage, recherche linéaire),
- échoue souvent pour les études « fortement » non-linéaires si elle est utilisée seule ; il apparaît nécessaire de la combiner avec une autre technique, comme la recherche linéaire.

Cet opérateur permet aussi d'extraire d'un calcul précédent la liste des instants réellement calculés et de la raffiner d'un facteur 2. Il est alors possible de relancer le même calcul avec cette nouvelle liste 2 fois plus fine afin de contrôler l'erreur due à la discrétisation temporelle.

3.2 Gestion des chargements

L'application d'un chargement en effort ou en déplacement imposé non-nul se fait usuellement de manière graduelle (par paliers de charge). Pour cela, il est possible de définir un chargement de type fonction (du temps), ou bien de multiplier le chargement par une fonction de type rampe (avec `FONC_MULT` sous le mot-clé `EXCIT` de `STAT_NON_LINE`), sauf en cas de pilotage.

Les charges servant aux conditions aux limites (de type déplacement imposé nul) ou au contact sont appliquées directement et ne nécessitent pas la définition d'une fonction pour une application graduelle.

3.3 Gestion de la convergence de l'algorithme global non-linéaire

Par défaut, l'algorithme de résolution globale non-linéaire est basé sur la méthode de Newton-Raphson. Le paramétrage de la convergence de l'algorithme de Newton est fait sous le mot-clé facteur CONVERGENCE de STAT_NON_LINE [U4.51.03].
La convergence de l'algorithme de Newton est caractérisée par la donnée d'une valeur seuil du résidu. Lorsque la valeur du résidu est dessous de ce seuil, l'algorithme a convergé. Par défaut, c'est le résidu relatif qui est pris en compte. Le seuil est alors donné par le mot-clé RESI_GLOB_RELA (10^{-6} par défaut). **Il est fortement déconseiller de modifier ce seuil de convergence**.
Lorsque le chargement imposé est nul (en cas de décharge totale par exemple), le résidu relatif ne fonctionne pas et on essaie de passer automatiquement à un critère de convergence absolu (RESI_GLOB_MAXI). Cette opération est transparente pour l'utilisateur. De manière générale il est déconseillé de renseigner RESI_GLOB_MAXI. Lorsque ces critères de convergence ne suffisent pas, il est possible d'utiliser des critères plus fins : un critère qui raisonne composante par composante (RESI_COMP_RELA) ou un critère qui raisonne à partir d'une valeur de référence pour chaque quantité (RESI_REFE_RELA).

3.4 Gestion de l'archivage

La question de l'archivage des champs calculés en implicite est une question moins importante qu'en dynamique explicite, où le nombre de pas de temps est généralement très élevé.

Suivant la taille de stockage dont on dispose, et le nombre de champs que l'on souhaite stocker, il est possible d'évaluer le nombre de pas temps que l'on peut stocker.
Par défaut, tous les pas de temps sont archivés, y compris les instants de calculs nouvellement créés par redécoupage automatique du pas de temps. Comme ce nombre ne peut être déterminé à l'avance, il est conseillé de n'archiver que les instants de la liste d'instants initiale. Si cette liste contient encore trop d'instants, il est conseillé de créer une 2ème liste, réduite, spécifique à l'archivage. Si la liste d'instants initiale contient beaucoup d'instants et que le sous-découpage automatique du pas de temps n'est pas activé (ce n'est pas conseillé), on peut aussi limiter l'archivage par une fréquence d'archivage.

Il est également possible de réduire l'encombrement de la structure de données resultat en extrayant certains champs (opérateur EXTR_RESU).

3.5 Observation et suivi de certaines grandeurs

Il est possible de stocker dans une table certaines composantes des champs calculés sur des parties du modèle à des instants d'une liste (dite d'observation) généralement plus fine que la liste des instants archivés. Cette fonctionnalité peut être activée dans STAT_NON_LINE sous le mot-clé OBSERVATION. Cela permet de réduire considérablement les besoins d'archivage. Cependant, il convient d'être prudent dans l'usage de l'observation. En effet, cette fonctionnalité a été conçue pour quelques points dans la structure et il n'est pas conseillé de faire plusieurs milliers d'observations à chaque pas de temps, pour des raisons de performance.

Il est également possible de suivre l'évolution de certaines composantes des champs calculés au cours des itérations de Newton et de les afficher dans le tableau de convergence. Cette fonctionnalité peut être activée dans STAT_NON_LINE sous le mot-clé SUIVI_DDL.

3.6 Affichage des informations et performance

De nombreuses informations sont affichées par défaut dans les fichiers message et résultats (.mess et .resu).

Une information essentielle est le tableau de convergence affiché à chaque instant de calcul. Ce tableau est très utile et permet notamment d'identifier certaines anomalies de convergence.

L'opérateur `STAT_NON_LINE` affiche la taille de la matrice (nombre d'équations du système).
Suivant la taille de la matrice, l'utilisation d'un solveur itératif (comme `PETSC`) permettra un gain en temps. À titre d'information, on considère qu'un solveur itératif est plus rapide qu'un solveur direct (par défaut) à partir d'environ 200 000 équations en 3d. La contre-partie de l'utilisation d'un solveur itératif est une robustesse non garantie. Il est à noter que l'utilisation de `PETSC` nécessite une version MPI de *Code_Aster* (même si on n'utilise qu'un seul processeur).

L'utilisation du parallélisme peut être aussi efficace si le temps passé dans `STAT_NON_LINE` est prépondérant par rapport au temps total du calcul. Pour cela, il suffit de choisir un solveur adéquat (exemple : `MUMPS` , `PETSC`), une version MPI de *Code_Aster* et de spécifier le nombre de processus dans Astk. De nombreux conseils sur le parallélisme sont donnés dans le document [U2.08.06]. Des informations plus fines sur les temps passés dans chaque partie de la résolution sont également affichées, et peuvent servir à mieux paramétrer le calcul (temps, mémoire, nombre de processeur...). On renvoie pour cela à la documentation [U1.03.03].

4 Non-linéarité matériau

Cette partie traite des problèmes les plus fréquemment rencontrés lors de l'utilisation de lois de comportements non-linéaires.

4.1 Paramètre liés à la résolution du comportement

La plupart des lois de comportement de *Code_Aster* s'intègre de manière analytique. En cas d'erreur dans l'intégration de la loi de comportement, il n'y a donc aucun paramètre numérique sur lequel on peut jouer. La solution usuelle consiste à re-découper le pas de temps global (ou à choisir une liste d'instants plus fine).
Certaines lois de comportement sont résolues localement via une méthode numérique itérative, pour laquelle l'utilisateur peut choisir le résidu maximum autorisé et le nombre d'itérations locales maximum. La plupart du temps, les paramètres par défaut (`RESI_INTE_RELA` = 10^{-6} et `ITER_INTE_MAXI` = 20) suffisent. En cas d'erreur dans l'intégration de la loi, on peut comme précédemment sous-découper le pas de temps, mais également autoriser des itérations locales en plus (`ITER_INTE_MAXI`). Par exemple, pour le `MONOCRISTAL`, il est fréquent d'autoriser 100 ou 200 itérations locales. Le choix du résidu est plus délicat. En effet, il est important que le comportement soit correctement intégré. Cela signifie qu'il est impératif de ne pas augmenter le résidu au-delà de la valeur par défaut (10^{-4} par exemple), sauf pour les comportements intégrés par un algorithme explicite. Par contre, il est parfois conseillé de resserrer le résidu (par exemple `RESI_INTE_RELA` = 10^{-10}). En fait, suivant la manière dont est implémenté le comportement, le résidu n'a pas la même signification physique (contrainte, déformation...). Il faut alors se référer à la documentation de référence de la loi de comportement en question.

4.2 Gestion de l'incompressibilité plastique

Si le matériau est incompressible $(\nu > 0.45)$ ou en cas de fortes déformations plastiques, des oscillations sur les contraintes ou sur la trace des contraintes peuvent apparaissent. Certains éléments finis spéciaux, comme les éléments sous-intégrés ou les éléments mixtes, permettent de traiter ces problèmes [U2.01.10]. Si l'utilisation d'éléments sous-intégrés ne règle pas le problème, alors on conseille l'utilisation de la formulation mixte à 2 champs (déplacement et pression) en petites déformations et de la formulation mixte à 3 champs en grandes déformations. À noter que cette dernière ne fonctionne que sur des maillages quadratiques.

4.3 Décharge

En cas de décharge, on peut rencontrer des problèmes de convergence de l'algorithme de Newton. On conseille d'utiliser la prédiction élastique (dans `STAT_NON_LINE` sous le mot-clé `NEWTON`). Il est

également utile d'activer le sous-découpage du pas de temps en cas de matrice singulière (c'est automatique en version 11, mais à activer à la main dans les versions précédentes *via* STOP_SINGULIER = 'DECOUPE' dans STAT_NON_LINE sous le mot-clé SOLVEUR).

4.4 Contraintes planes

Le traitement de la condition de contraintes planes est réalisé dans le cadre général par la méthode de Deborst. Il est conseillé de réactualiser souvent la matrice tangente (toutes les une à trois itérations de Newton). Dans certains cas, la convergence est atteinte pour l'algorithme de Newton, mais pas pour la vérification de l'état de contraintes planes, ce qui conduit à des itérations supplémentaires, voire un re-découpage excessif du pas de temps. Il est alors conseillé d'activer une boucle supplémentaire pour mieux satisfaire les contraintes planes au cours des itérations de Newton : ITER_CPLAN_MAXI doit être choisi au moins égal à 5 [U4.51.11].

4.5 Endommagement (problèmes adoucissants)

Pour traiter les problèmes d'endommagement, de nombreux conseils sont donnés dans [U2.05.06]. De manière générale, on conseille d'utiliser le pilotage, avec éventuellement la recherche linéaire mixte.

Pour certaines lois de comportement (ENDO_FRAGILE et ENDO_ISOT_BETON), la méthode IMPLEX est proposée en alternative de la méthode de Newton. On active cette méthode dans STAT_NON_LINE sous le mot-clé NEWTON. Cette méthode se base sur une extrapolation explicite des variables internes pour déterminer les déplacements à partir desquels le comportement est intégré implicitement. La nullité de l'équilibre n'est pas vérifié. De ce fait, elle introduit une approximation de la résolution mais permet de garantir la robustesse du calcul. Un contrôle de l'erreur et une optimisation des pas de temps sont possibles via l'opérateur DEFI_LIST_INST en choisissant une gestion automatique du pas de temps et un mode de calcul des pas de temps spécifique à IMPLEX. D'un point de vue pratique, la méthode impose une ré-actualisation de la matrice à chaque incrément et une seule itération. Avant toute utilisation de la méthode IMPLEX, le document [R5.03.81] doit impérativement être consulté.

4.6 Thermo-Hydro-Mécanique (THM)

Les problèmes de THM font intervenir des notions bien particulières. On conseille de consulter la documentation [U2.04.05]. On donne ici quelques conseils généraux pour la résolution. L'initialisation de champs est une étape délicate à laquelle il faut bien faire attention (ETAT_INIT). Il est nécessaire d'utiliser la matrice tangente ré-actualisée. En cas de problème de convergence, il peut être très utile d'activer la recherche linéaire (mixte de préférence). La recherche linéaire n'améliore cependant pas systématiquement la convergence, elle est donc à manier avec précaution dans la mesure où elle peut accroitre le coût CPU.

5 Non-linéarité géométrique

Cette partie traite des problèmes les plus fréquemment rencontrés lors de calculs avec des non-linéarités géométriques.

5.1 Flambage

L'opérateur STAT_NON_LINE permet de calculer un critère de stabilité *via* le mot-clé CRIT_STAB=_F (TYPE='FLAMBEMENT'). La documentation d'utilisation du flambage [U2.08.04] donne de nombreux conseils sur la mise en place d'un calcul de stabilité. On donne ici quelques conseils généraux.
De manière générale, l'analyse de stabilité est conduite sur matrices de raideurs réactualisées. En pratique, il faut limiter les appels à CRIT_STAB pour des raisons de coûts de calcul, en limitant les instants sur lesquels l'analyse de stabilité est réalisée. En complément, il est judicieux de n'utiliser CRIT_STAB que sur les intervalles de temps où l'on soupçonne la possibilité d'instabilités. Il convient également de bien raffiner le pas de temps à l'approche de cette zone. Par défaut, on calcule 3 charges critiques (NB_FREQ). Souvent la première peut suffire. Un des points particuliers liés à l'instabilité est le choix de la technique de pilotage de l'algorithme. En effet, le pilotage classique en

effort n'est pas adapté. À l'approche d'un point limite, il faut réduire l'incrément de charge et augmenter le nombre maximal d'itérations. Il est également conseillé d'utiliser le pilotage par longueur d'arc.

5.2 Grandes déformations

Le mot-clé DEFORMATION sous COMPORTEMENT permet de définir les hypothèses utilisées pour le calcul des déformations. Par défaut, on considère de petits déplacements et petites déformations. (DEFORMATION = 'PETIT'). Cela signifie que l'on reste en Hypothèse des Petites Perturbations : petits déplacements, petites rotations, petites déformations (inférieures à environ 5%). Lorsque cette hypothèse n'est plus vérifiée, il faut changer de modèle de déformations. Pour les structures élancées (coques, plaques, poutres), il arrive fréquemment que l'on soit en grands déplacements, grandes rotations mais petites déformations. On utilise alors DEFORMATION = 'GROT_GDEP'.

Le traitement des grandes déformations diffère suivant le type d'élément et la loi de comportement. Le modèle de grandes déformations de Simo et Miehe (DEFORMATION = 'SIMO_MIEHE') est conseillé pour les relations de comportement VMIS_ISOT_LINE, VMIS_ISOT_TRAC, ROUSSELIER et tous les comportements à écrouissage isotrope uniquement, associés à un matériau subissant des changements de phases métallurgiques (relations META_X_IL_XXX_XXX et META_X_INL_XXX_XXX). Pour les autres lois de comportement, on conseille le modèle de grandes déformations de Miehe et Apel (DEFORMATION = 'GDEF_LOG'), théoriquement applicable à n'importe quelle loi de comportement pour les modélisations 3D et 2D. Des explications plus détaillées se trouvent dans la notice d'utilisation des comportements non-linéaires [U4.51.11] au paragraphe DEFORMATION.

6 Non-linéarité de contact-frottant

La définition de la charge de contact-frottant fait l'objet d'un opérateur spécifique : DEFI_CONTACT [U4.44.11]. Outre le choix des entités en contact, cet opérateur permet de choisir le type de formulation du problème de contact (discrète ou continue), des paramètres pour l'opération d'appariement et des paramètres pour la phase de résolution. De nombreux conseils de mise en données sont disponibles dans le document [U2.04.04], notamment sur le choix des surfaces maître et esclave.

La prise en compte des conditions de contact avec ou sans frottement conduit à des problèmes non-linéaires, qui peuvent être source de difficultés dans la résolution de STAT_NON_LINE. Une des premières difficultés est le blocage des modes de corps rigides. En effet les conditions de contact-frottant n'interviennent pas dans le bocage des modes rigides. Il faut donc bloquer les modes rigides sans tenir compte des conditions de contact (sauf de rares exceptions). Un mauvais blocage des modes de corps rigides entraîne l'apparition de pivots nuls lors de la factorisation de la matrice. Des conseils sont donnés dans la documentation [U2.04.04]. Attention, certaines formulations de contact nécessitent un solveur particulier. Notamment, en formulation discrète, la méthode CONTRAINTE nécessite un solveur direct (MULT_FRONT ou MUMPS). De manière générale, on déconseille l'utilisation de la recherche linéaire avec le contact. En cas de couplage avec d'autres non-linéarités, il est conseillé de s'assurer que chaque non-linéarité prise séparément converge. Si des problèmes n'apparaissent que lorsque les non-linéarités sont couplées, il est conseillé de travailler avec la matrice tangente réactualisée et une prédiction élastique.

Un benchmark comparatif entre *Code_Aster* et d'autres logiciels commerciaux a été réalisé en 2010 sur 5 problèmes de contact-frottant [2]. Ce benchmark a notamment permis de mettre en évidence des oscillations de la pression de contact pour la formulation continue en cas de maillages incompatibles, défaut dont souffrent aussi les logiciels commerciaux utilisés. Il est alors recommandé d'extraire les pressions de contact à partir des contraintes sur le bord, ou bien d'utiliser des maillages compatibles.

7 Bibliographie

[1] S. Geniaut, Premier REX sur la gestion automatique du pas de temps dans STAT_NON_LINE, CR-AMA-09.268, 2009

[2] T. De Soza, Bilan du lot « Contact/Frottement » en 2010 (projet MNAM), CR-AMA-11.049, 2011